Progress Report 30 July 2001 Dear CellML Guy — More Questions About CellML

Author:

Warren Hedley (Bioengineering Institute, University of Auckland)

1 Introduction

This document continues the <u>19 June 2001 Meeting Minutes</u>¹, in which I endeavoured to answer some questions from readers of the <u>18 May 2001 Final Draft of the CellML specification</u>². These are intended to be candidate Q&A's for the CellML FAQ, which I'm sure will be updated at some point.

2 What CellML elements are re-usable?

An anonymous reader writes:

Dear CellML guy,

Based on my reading, the only re-usable unit as a submodel is a **<component>**, and a hierarchy of components can only be defined by using the **<group>** element, but (p47, bottom) a **<group>** element "must be a child of a **<model>** element." That means only the most atomic components can be re-used as components! I showed this to other people here and they assumed that this was just a typo, and it should say, "**<group>** must be a child of a **<model>** or **<component>**". If this is simply a typo, please fix. If it is not, then it seems like a major limitation to re-usability. For example, in reaction networks, each reaction should belong to its own component according to the spec. Now imagine the simple GPCR signaling motif:

Gabg --(R-bound)--> Ga-GTP + Gbg --(AC)--> Ga-GDP + Gbg ----> Gabg

If I understood the spec correctly, you would not be able to store something as simple as this as a re-usable component without violating "good practices" which would require each reaction to be its own component. Please clarify!

The CellML Guy replies:

First of all, a general observation: a common misconception among those who don't deal on a day to day basis with XML is that "re-use" means "must be cut-and-pastable". In fact, this is not what re-usable means when you see it in the CellML specification and associated documentation. CellML re-use occurs at the abstract data model level. The authors of the CellML specification have invested considerable effort in making sure that the basic "objects" in the CellML data model can be re-used and re-combined with the minimum of fuss — however, the re-use occurs at the software level, not at the XML level. I anticipate that model authoring and editing software will allow users to re-use and manipulate components representing physiological regions, species or reactions in a straightforward manner. When the user chooses to export the model as CellML, the result may bear no resemblance to the CellML descriptions of the components from which it was created.

To address your point regarding the valid placement of the **<group>** element: it is important to remember that **the CellML data model is based on a network of connected components, over which several hierarchies may be defined**. One of these, the *encapsulation* hierarchy, restricts how the components in the network may be connected. Other types of hierarchy involve no such restrictions, but allow the modeller to group components together, and arrange them hierarchically. Because a CellML **<model>**

¹http://www.cellml.org/private/progress_reports/20010619_meeting_minutes.html

²http://www.cellml.org/public/specifi cation/20010518/index.html

consists of a network, the component elements are simply listed in sequence within the **model>** element. Specifically, they can not be nested within each other to define pseudo-hierarchical arrangements. It would, as you suggest, be possible to define a hierarchical relationship between a given component and its children within the **component>** element using a **group>** element. However, as is discussed in <u>Section</u> 5.4 (Containment and Encapsulation) of the Hodkgin Huxley Squid Axon model documentation³ the recommended best practice for defining hierarchical arrangements of components is to define each hierarchy within its own **group>** element, allowing people and software looking at or interpreting the XML code directly to read the entirety of each hierarchy in one hit. It thus makes more sense to define **group>** elements within **model>** elements. Allowing **group>** elements to be defined in **component>** elements would introduce unnecessary complexity into the CellML specification.

I am not familiar with the simple GPCR signalling motif you mention, but imagine that you might want to take a look at the <u>Simple Two Reaction Model With Encapsulation</u>⁴ example on the CellML website. In that example two reactions and supposedly irrelevant intermediate byproducts are encapsulated within a single reaction component that represents the total process. This reflects the recommended best practices for defining pathway models in CellML: use a single component to represent each species and reaction, and use encapsulation to hide what you consider to be unnecessary detail when appropriate.

3 What is the conceptual difference between the <model> and <component elements?

The anonymous reader continues:

Dear CellML guy,

I do not understand the distinction between <model> and <component>. Would it not be possible (if you allowed <group> to be the child of a <component>, and then also allow the component to specify its own default values for variables declared with public or private interfaces of "in") to have a component be a free-standing runnable model, but in which the default values for "in" are superseded if a variable mapping is established to that variable, i.e., if the component is then used in a larger model?

The CellML Guy replies:

This question was partly answered in my reply above, and can be pretty much summed up in the following sentence: a model is a network of connected components. CellML components should correspond to the smallest functional units in a model. A model consists of a set of these functional units, and the connections between them. This, by itself, doesn't mean much. A more practical consideration is that of component and unit namespaces. The identifiers of **<component>** and **<unit>** elements must be unique across a model. If **<component>** elements could be nested, it could get very confusing, with multiple components with the same name within a model.

The <u>Meeting Minutes from 31 October 2000</u>⁵ proposed a scheme for combining models into larger models, based on the <u>Low-Level XML Re-use Scheme</u>⁶ developed in 2000. This involved the nesting of model elements. A parent <model> element could create connections between components in submodels by using the model's identifiers as namespaces for the submodel's components. A simplified version of the example from the 31 October 2000 Meeting Minutes is shown in Figure 1. Note that the combination of models in this way would still be consistent with the idea that a model was a network. The combination of models was never considered further because of its dependence on the Low-Level XML Re-use Scheme, which is way ahead of its time.

³http://www.cellml.org/examples/examples/electrophysiological_models/hh_squid_axon_1952/index.html#sec_containment_and_encapsu-⁴http://www.cellml.org/examples/examples/signal_transduction_models/basic_reaction_models/two_reaction_model_with_encapsulation.

⁵http://www.cellml.org/private/progress_reports/20001031_meeting_minutes.html

⁶http://www.cellml.org/private/documentation/component_reuse.html

```
<model name="combined_model">
<model name="my_electrophysiological_model">
    ....
    </model>
    <model name="my_mechanical_model">
    ....
    </model>
    <model name="my_mechanical_model">
    ....
    </model>
    <model>
    <model>
    <models
    <monopenents
        components
        component_1="my_electrophysiological_model.intra"
        component_2="my_electrophysiological_model.intra" />
        <map_variables variable_1="calcium" variable_2="calcium" />
        </connection>
    <//r>
```

```
</model>
```

FIGURE 1: A method for combining models into larger models proposed in the Meeting Minutes from 31 October 2000.

4 How can I add information about reaction types (e.g., "Michaelis Menten") to my CellML document?

The anonymous reader spouts forth:

Dear CellML guy,

Could you outline how the "type" (e.g., "michaelis menten", "mass action") of a reaction might be specified by a CellML writer so that a compatible reader could read in the reaction and recognise that it belongs to one of several pre-defined reaction types? Is this encodable as metadata? As a special extension to cellML? I didn't see anything about this in the reaction section of the spec. I bring this up for two reasons: (1) might be useful for interchangeability with SBML, (2) so that software didn't have to try and reverse reaction type out of reaction mathematics, which sounds tricky, and in fact impossible in some cases, where "flux" and "mass action" might be identical mathematically.

The CellML Guy replies:

As you correctly surmised, reaction type information would be considered metadata for a CellML document. As far as CellML is concerned, model data consists of structure and mathematics, although we've allowed some reaction participant classification stuff to get in, because apparently biologists think that stuff is important. Eventually there will be three ways to embed such information in a CellML document:

- Information can be embedded as metadata using the framework described in <u>Section 8 of the CellML</u> <u>specification</u>⁷. The <u>CellML Metadata specification</u>⁸ will undoubtedly propose some standard way of describing reaction type, among other things.
- Various biological ontology standards are currently being developed, and future versions of CellML will undoubtedly draw on these, providing a shorthand for associating CellML objects with biological ontologies.

⁷http://www.cellml.org/public/specifi cation/20010518/metadata.html ⁸http://www.cellml.org/public/metadata/20010518/index.html

⁸http://www.cellml.org/public/metadata/20010518/index.html

• Anything not covered in the main specifications can always be embedded as application-specific extension data.

E-mail questions, criticism, submissions or info to info@cellml.org Input document last modifi ed : Mon Feb 02 15:25:02 NZDT 2004